Cue Integration and Discrete MRFs towards Knowledge-based Segmentation and Tracking

Ahmed Besbes, Nikos Paragios, Nikos Komodakis

▶ **To cite this version:**

Ahmed Besbes, Nikos Paragios, Nikos Komodakis. Cue Integration and Discrete MRFs towards Knowledge-based Segmentation and Tracking. [Research Report] RR-6831, INRIA. 2009, pp.24. inria-00359612

HAL Id: inria-00359612
https://hal.inria.fr/inria-00359612

Submitted on 8 Feb 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Cue Integration and Discrete MRFs towards Knowledge-based Segmentation and Tracking

Ahmed Besbes — Nikos Paragios — Nikos Komodakis

N° 6831

February 2008

Thème BIO

ISSN 0249-6399 ISRN INRIA/RR--6831--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
SACLAY - ÎLE-DE-FRANCE

Cue Integration and Discrete MRFs towards Knowledge-based Segmentation and Tracking

Ahmed Besbes*† , Nikos Paragios*†, Nikos Komodakis‡

Thème BIO — Systèmes biologiques
Équipe-Projet GALEN

Rapport de recherche n° 6831 — February 2008 — 24 pages

Abstract: In this report, we propose a novel similarity-invariant approach to model shapes. The method assumes a control points representation of the model and an arbitrary interpolation strategy. First, we construct the prior manifold using the distributions of the relative normalized distances between pairs of control points within the training set. The considered shape model refers to an incomplete graph that consists of intra and inter-cluster connections representing the inter-dependencies of control points. The clusters are determined according to the co-dependencies of the deformations of the control points within the training set. Then, we introduce a geometric partition of the space using a Voronoi decomposition that aims to determine relationships between the control points and the image domain. The same prior model is extended to the temporal domain to encode dynamic dependencies between the control points in the case of image sequences. We apply our model to both segmentation and tracking. Our knowledge-based approach to solve these problems is expressed using a Markov Random Field, where the unknowns are the positions of the control points. The pairwise potentials encode the variations of the shape, while the singleton potentials refer to the data term through the Voronoi decomposition of the space, and to the dynamic constraints. State-of-the art techniques from linear programming are considered both for the clustering and the optimization of the designed function. We present our results for the segmentation of the hand and the left ventricle in CT images, and the tracking of walking people.

Key-words: Segmentation, Tracking, Shape Modeling, Left Ventricle, MRFs

We thank Georg Langs from the Medical University of Vienna for his contribution with the shape maps code, Xiang Zeng and Dimitris Samaras from the Stony Brook University for their contribution to the tracking algorithm, and Radhouène Neji from GALEN for the interesting discussions we had and the suggestions he gave particularly for the graph construction.

* Laboratoire MAS, Ecole Centrale Paris, Châtenay-Malabry, France.
† Equipe GALEN, INRIA Saclay - Île-de-France, Orsay, France.
‡ Department of Computer Science,University of Crete, Greece.

Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université
4, rue Jacques Monod, 91893 ORSAY Cedex
Téléphone : +33 1 72 92 59 00

Intégration d'indices et Champs de Markov Discrets pour la Segmentation et le Suivi de Mouvement

Résumé : Dans ce rapport, nous proposons une nouvelle approche invariante par similitudes pour la modélisation de formes. Notre méthode suppose une représentation du modèle basée sur des points de contrôle, et une stratégie d'interpolation arbitraire. D'abord, nous construisons la variété a priori en utilisant les distributions des distances normalisées entre paires de points de contrôle dans l'ensemble d'apprentissage. Le modèle de forme considéré correspond à un graphe incomplet constitué de connections intra et inter-groupes qui représentent les dépendances relatives des points de contrôle. Ces groupes sont déterminés via les dépendances relatives des déformations des points de contrôle dans l'ensemble d'apprentissage. Ensuite, nous partitionnons l'epace grâce à une décomposition de Voronoi pour déterminer des relations d'appartenance entre les points de contrôle et les pixels (ou voxels) de l'image. Le modèle d'intéractions saptiales est étendu au domaine temporel pour représenter les dépendances dynamiques entre les points de contrôle dans le cas de séquences d'images. Nous appliquons notre modèle aux problèmes de segmentation et de suivi de mouvement. Notre approche, qui utilise des informations a priori, est exprimée à travers un champ de Markov, avec comme inconnues les positions des points de contrôle. Les potentiels "doubles" contraignent la forme à rester dans l'espace appris, alors que les potentiels "simples" lient le modèle aux données à travers le diagramme de Voronoi d'un côté, et expriment les contraintes dynamiques de l'autre. Des techniques récentes et efficaces de programmation linéaire sont considérées pour extraire les groupes de points et pour minimiser l'énergie du graphe. Nous présentons nos résultats pour la segmentation de la main et du ventricule gauche dans les images CT, et pour le suivi de mouvement de personnes.

Mots-clés : Segmentation, Suivi de Mouvement, Modélisation de formes, Ventricule Gauche, Champs de Markov

Contents

1 Introduction

1.1 Context and Motivation

Shape modeling and its application to knowledge-based object segmentation and tracking are fundamental tasks of computer vision. In fields where a prior knowledge is available (like medical imaging), such a method carries on great potentials since the domain knowledge can be used to introduce constraints which improves the final reliability and accuracy of the segmentation or the tracking result. In order to do so, one first has to determine a model representing these constraints and then an inference process which aims to combine the visual support with the prior knowledge.

1.2 Previous Work

1.2.1 Shape Modeling and Segmentation

The definition of the shape model involves a representation and a statistical model. Landmark-based representations are widely used in computer vision. Point-based representations [12] are a typical example of such methods that have been studied widely in the context of active contours and snakes [38] through continuous interpolation strategies. Implicit representations [30] are an alternative approach to model shapes that handle topological changes naturally, while being computationally inefficient. The above-mentioned methods reconstruct the shape through local or global interpolation.

Once the representation has been considered, the next task consists of modeling its variations within a training set in order to construct the prior. In this context, simple average models [7], principal component analysis [12], as well as their kernel variants [14], Gaussian densities [32], mixture models, and non-parametric priors were considered. These methods make an explicit assumption on the nature of the statistical behavior of the training set and then determine the optimal set of parameters towards representing the observed variations.

Image-based inference is the last issue to be addressed where one aims to combine the visual support with the prior model. To this end, a cost function that combines both edge-based as well as region-driven terms is often considered. The main challenge is to determine the corresponding optimal solution that is often challenging with gradient-based approaches [38]. On the other hand, discrete methods [6] could yield a better minimum of the objective function under certain constraints but the integration of global deformable priors [17][33] is not straightforward.

As far as the cardiac segmentation case is concerned, both model-based and model-free approaches were applied. The afore-mentioned active shape models [12], active appearance models [11] and their variants have been particularly used in the segmentation of the left ventricle [1][29]. Snakes, active contours [28] and their level set variants [31] have also been applied to the segmentation of the left ventricle. Discrete optimization and Markov Random Fields (MRFs) [18] are another class of methods that have gained significant attention in the last years. This is mostly due to their ability to capture a better minimum of the objective function under certain constraints. These methods have been used to address segmentation at the pixel level [5][19] or introduce some notion of global prior in terms of shape geometry [17]. However, the main limitation of these methods is their inability to cope with complex interactions between pixels (graph nodes) which is often the most prominent way to introduce global deformable

prior models at a reasonable computational complexity. The reader is referred to [16] for a thorough survey of 3D cardiac segmentation methods.

1.2.2 Object Tracking

Prior work on tracking involves blob-based appearance methods, static shape and appearance priors as well as dynamical systems.

In the first case, one can cite for example methods like the Lucas-Kanade tracker [27], correlation-based methods [34], the mean-shift [10] and its numerous variants [9]. These methods use a similarity criterion to compare the previous appearance of the object with possible candidates in the new images towards recovering the most probable position. Different search strategies are used to recover the most probable position. These methods are a good compromise in terms of performance and computational complexity but cannot cope with deformations.

Shape and appearance priors aim to represent statistically the variations of the class of objects being tracked. This manifold is then used often in conjunction with a maximum a posteriori (MAP) estimation and the image towards recovering the object and its deformations in a new image. Active shape models [12], active appearance [11], level set methods with priors [32] are some examples of these methods. They perform well in linear subspaces, but fail to account for highly deformable objects. In the most general case, blob-based methods as well as static priors do not use prior tracking results to impose consistency.

Dynamical systems are a promising alternative to encode tracking dynamics. One can site numerous examples using like Kalman filtering [22], condensation [21], or multiple hypotheses testing and particle filter tracking [2]. These methods perform better than blob-based methods while recovering an explicit model to represent the temporal motion of the object. Due to computational complexity constraints, the integration of these methods with deformable tracking was not so successful.

One should also mention methods aiming to track articulate models. In such a context, objects are presented with parts, and then constraints between the relative positions of these parts are introduced [15][3][40][42][23]. These methods can be very efficient but assume an explicit hierarchical representation of the model, and impose constraints on their dynamics. Therefore, their use in the context of tracking arbitrary highly deformable objects is not adequate.

1.3 Contributions of this Work

We propose a novel approach to knowledge-based segmentation and tracking using efficient linear programming. To this end, we extend the chord length distribution representation (CLD) [37] and we introduce a novel invariant shape representation (with respect to translation, rotation and scale), that defines the manifold using pairs of points and the probability density distributions of their relative normalized distances. These densities are learned from the training set through simple statistical inference. Such representation encodes global shape consistency through local shape interactions. These control points are clustered according to their behavior within the training set using a linear programming approach [24]. Clusters correspond to sets of points for which one can predict with certain confidence the positions given the position of the cluster center. This clustering is obtained through an inference process that measures the statistical similarity in terms of deformation between pairs of control points using

shape maps [26]. On the other hand, the relative positions of cluster centers with respect to control points that do not belong to their clusters encode the global structure of the shape. Then, we model the shape variation through probability densities that encode the aforementioned local and global dependencies. The resulting framework can encode simple or complex distribution models according to the entropy of the observed system, unlike [20] where the model is assumed to be Gaussian. Such a model is represented using an incomplete graph having a k-fan structure [13] derived from the training set, where the connections between the components of a cluster represent the local dependencies, while the connections between the clusters centers account for the global correlations between parts of the shape. Similarly, temporal priors are encoded through relative deformations of different control points in time once motion has been implicitly accounted for.

Then, inference consists of deforming the model consistently with the image information. The unknown parameters refer to the positions of the control points. In order, to determine the support from the image, we propose a Voronoi decomposition of the domain, defining a membership function that relates the pixels to the control points. The data term is then determined using this decomposition, while the prior term is expressed using the pairwise potentials between control points. Recent advances in the area of discrete optimization which explore the duality theorem of linear programing [25] are considered to recover the lowest potential of the objective function on one hand, and the clusters on the other hand [24]. Shape modeling in 2D and 3D, the segmentation of the hand and the left ventricle, and the tracking of walking people and faces are the applications being considered to demonstrate the potentials of our approach.

2 Shape Representation

Knowledge-based segmentation and tracking methods are based on the definition of a model which is then combined with image measurements towards object extraction. Classic approaches consist of representing the shape using a number of landmarks and learning their behavior using a training set.

2.1 A Point Distribution Model

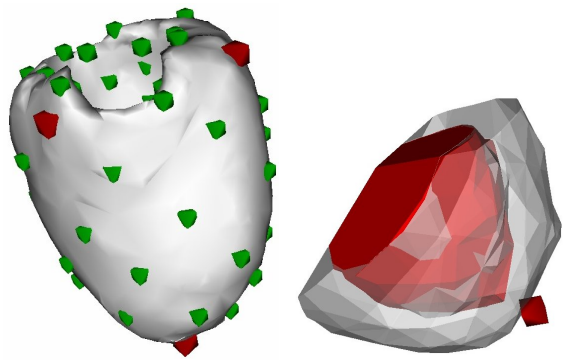
Our shape model $\mathbf{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ consists of a set of n control points lying on its boundary ([Fig. 1(a)]). The contour of the shape can be recovered for example by interpolating the positions of these control points. The information carried by our model is described in a similarity-invariant manner, extending the CLD representation [37], and using the distances $\mathbf{d}_{ij}$ between pairs of control points $(\mathbf{x}_i, \mathbf{x}_j)$, normalized by the scale $\mathbf{d}$ of the object, or:

$$\mathbf{d}_{ij} = \frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\mathbf{d}} \quad , \tag{1}$$

where $\mathbf{d} = \frac{2}{n.(n-1)} \sum_{i=1}^{n} \sum_{j>i}^{n} \|\mathbf{x}_i - \mathbf{x}_j\|$. Let us consider now a set $\mathcal{S} = \{s_1, \ldots, s_m\}$ of m instances of the object, where each example is represented using n control points, i.e. $s_u = \{x_1^u, \ldots, x_n^u\}, \forall u \in \{1, \ldots, m\}$. Hence, $\forall i \in \{1, \ldots, n\}$, the set $\mathcal{X}_i = \{x_i^1, ..., x_i^m\}$ represents instances of the i^{th} control point of the shape. In practice, this training set is obtained by manually labeling the landmarks for each instance of the

shape, or by deducing the landmarks from the registration between a labeled shape and a set of non-labeled shapes.

Then, given a statistical model, we learn from the training set the probability density distributions of the relative positions of the control points $p_{ij} \equiv p(\mathbf{x}_i, \mathbf{x}_j) \equiv p(\mathbf{d}_{ij})$. These $\frac{n.(n-1)}{2}$ distributions are usually learned from a training set of instances of the object, where the control points are located in a consistent manner that guarantees correspondences. While, the constraints imposed by each pair are weak (given the position of a control point, the second should live in a circle with the learned distance as radius), the accumulation of them, through all possible pair interactions, could quite well approximate the manifold. The use of such model encodes global dependencies, while being expressed as local combinations of individual densities. Furthermore, such a model can account for local variations of varying complexity due to the fact that different statistical models can be used to express the pairwise densities towards capturing the observed variations. However, this representation can suffer from redundancy and can be expensive in terms of computation cost. Defining the most sparse set of pairs that represents best the shape is an interesting problem that we tackle in the following.

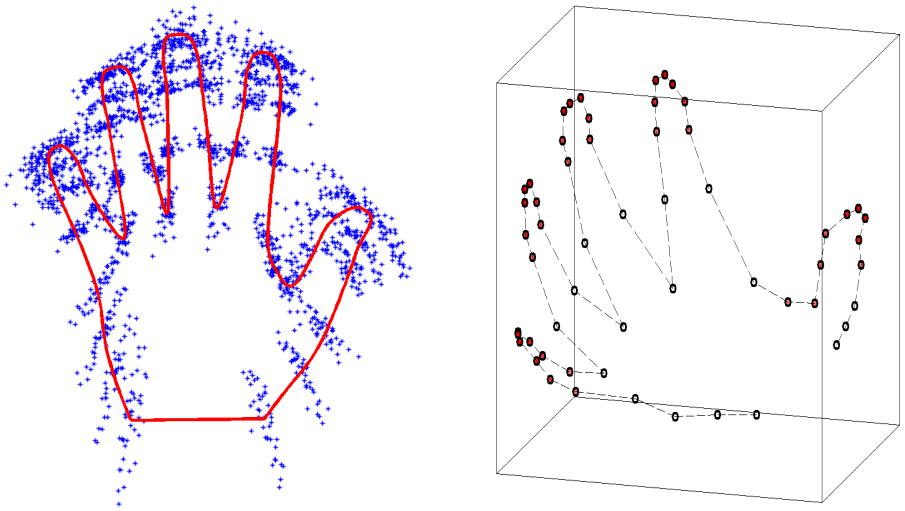


(a) (b)

Figure 1: Our model: a deformable shape associated with control points. (a) In red: control points P_4(in the apical and basal parts) used to define affine transforms of the shape. In green: the set of control points P_{90} that defines the TPS deformation. (b) The apical control point with the associated Voronoi cell, intersected with the blood pool and the myocardium.

2.2 Removing Redundancy

The task of eliminating the redundancy in the model, while preserving its ability to represent the data, is related to the minimum description length principle on one hand, and can be thought as a spectral clustering problem on the other hand. We aim to obtain as compact a model as possible assuming that the high dimensional data space can be approximated by a lower dimensional embedded manifold, which reduces the dimension of the problem significantly. Shape maps [26] handle precisely these two aspects, and are learned from the data in a way closely related to the diffusion maps [8], but using the compactness of models that describe sub sets of the entire data instead of the spatial distances or similarities between individual points. Therefore, we compute



(a) (b)

Figure 2: Model construction. (a) Aligning the training set using Procrustes Analysis. (b) The density or redundancy is color coded on the projection of the control points in the first 3 shape map dimensions. It reflects the coherence of local shape variation.

the shape map of the control points, using the training set $\{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$, and then we cluster the control points according to their mutual shape map distances. For a pair of control points $(\mathbf{x}_i, \mathbf{x}_j)$, the resulting map distance will be noted $\mathbf{d}_{sm}(\mathbf{x}_i, \mathbf{x}_j)$. For the hand example, the obtained redundancy is color-coded in [Fig. 2(b)]. A new clustering algorithm [24] was used for this final task, and is described in the section 2.3. The obtained clusters reflect the interdependencies between the control points, and refer to the parts of the object that have highly-correlated relative displacements.

2.3 Clustering via Linear Programming

Clustering refers to the process of organizing a set of objects into groups, where the members of each group are as similar to each other as possible. More formally, a common definition for clustering is the following one: suppose we are given a set of objects $\mathcal{V} = \{v_1, \ldots, v_n\}$ endowed with a distance function d that can measure the similarity between any two objects $v_i, v_j \in \mathcal{V}$. In such a case, the goal of clustering is to choose K objects, say, $\{c_1, c_2, \ldots, c_K\}$ from $\mathcal{V}$ (these will be referred to as the *clusters centers*), so that the obtained sum of distances between each object and its closest center is minimized, or:

$$\min_{c_1, c_2, \ldots, c_K} \sum_{v_i \in \mathcal{V}} \min_{c_k} d(v_i, c_k) \quad . \tag{2}$$

A common drawback of many popular clustering techniques (such as the K-means algorithm) is that they need to be given the number K of clusters beforehand. However, this is problematic as this number is very often not known in advance. To address this issue, we will let this number vary as well and change the objective function of clustering so as to assign a penalty (denoted by $d(v_i, v_i)$) whenever an object v_i is

chosen as a cluster center, or:

$$\min_{K,c_1,c_2,\ldots,c_K} \sum_{v_i \in \mathcal{V}} \min_{c_k} d(v_i, c_k) + \sum_{c_k} d(c_k, c_k) \ . \tag{3}$$

Another very bad symptom of many clustering techniques is that they are particularly sensitive to initialization. For instance, the K-means algorithm (which is one of the most commonly used clustering techniques) is doomed to fail if its initial cluster centers happen not to be near the actual cluster centers. To address this very important issue, we have used a novel clustering method. The main idea behind our method is to first formulate the clustering as a linear integer program as follows:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} d(v_i, v_j) x_{ij} \tag{4}$$

$$\text{s.t.} \sum_{j=1}^{n} x_{ij} = 1, \ \forall i \tag{5}$$

$$x_{ij} \leq x_{jj}, \ \forall i \neq j \tag{6}$$

$$x_{ij} \in \{0, 1\}, \ \forall i, j \tag{7}$$

In the above formulation, the binary variable x_{ij} (with $i \neq j$) indicates whether object v_i has been assigned to cluster center v_j or not, while the binary variable x_{jj} indicates whether object v_j has been chosen as a cluster center or not. It is then very easy to prove that the above linear integer program is actually equivalent to minimizing the objective function (3) of clustering. To this end, it suffices to observe that (5) simply expresses the fact that each object v_i can be assigned to exactly one cluster center v_j, while (6) simply expresses the fact that if any object v_i has been assigned to an object v_j, then v_j must be chosen as cluster center. To obtain an approximately optimal solution to the above integer program, we will then rely on first solving its linear programming relaxation and then "rounding" the relaxed solution in an appropriate manner. More details about the formulation of the problem and its optimization are given in [24]. In the validation section of [24], it is stated that a constant penalty cost, roughly set to the median of the distances $d(v_i, v_j)$ is used in the experiments . We also considered the same penalty value for our tests.

In our case, the set of objects correspond to the control points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, and the distance d corresponds to the aforementioned distance $\mathbf{d}_{sm}$ in section 2.2. We give in [Fig. 3(a)] an example of the output of this clustering process using the hand database [35].

2.4 The Shape Model

Before proceeding, let us summarize the model we obtain after the clustering step. Our shape model $\mathbf{M} = (\mathbf{S}, \mathbf{P})$ is an incomplete graph. It consists of a set of control points (unary cliques) $\mathbf{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ lying on the boundary of the object, and a set $\mathbf{P} = \mathbf{P}_l \cup \mathbf{P}_g$ of pairs of control points (binary cliques), where $\mathbf{P}_l$ contains the "local" pairs, and $\mathbf{P}_g$ contains the "global" pairs, or:

$$(i, j) \in \mathbf{P}_l \iff \mathbf{x}_i \in C(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in C(\mathbf{x}_i) \tag{8}$$

$$(i, j) \in \mathbf{P}_l \iff \mathbf{x}_i \notin C(\mathbf{x}_j) \text{ or } \mathbf{x}_j \notin C(\mathbf{x}_i) \tag{9}$$

(a) (b) (c) (d) (e) (f)

Figure 3: Model representation. (a) Control points clustered in 11 clusters: centers are represented by squares. (b)–(e) Deformation of a point cloud according to the shape prior term. (f) Voronoi decomposition of the model domain.

where $C(\mathbf{x}_i)$ is the cluster having $\mathbf{x}_i$ as center. Hence, each cluster center is connected to the control points in its cluster (local pairs) and to all the other control points (global pairs), which leads to a k-fan graph structure [13]. The novelty here consists in the method that defines automatically from the training data the number of fans and their centers. To each one of these pairs $(\mathbf{x}_i, \mathbf{x}_j)$ we associate a probability density distribution p_{ij} learned from the training set as previously stated in section 2.1. In practice, applying shape prior constraints to an initial set of random control points leads to an instance of the learned object, as showed in [Fig. 3(b)–3(e)]. The use of such prior in the segmentation framework is a much more interesting application and it is explored in the following.

2.5 The Dynamic Model

We extend the formulation of the shape prior to the temporal domain, in the case of image sequences, to define a dynamic prior. We introduce here the notation $\mathbf{x}_i^t$ to refer to the i^{th} control point at the frame t. Let us consider the temporal distance between two landmarks in different frames separated by an index τ, with global motion being accounted for through translation of the gravity center:

$$\mathbf{d}_{ij}^{t,t-\tau} = \frac{\left\|(\mathbf{x}_i^t - \bar{\mathbf{x}}^t) - (x_j^{t-\tau} - \bar{\mathbf{x}}^{t-\tau})\right\|}{\frac{1}{n^2}\sum_{k=1}^{n}\sum_{l=1}^{n}\left\|(\mathbf{x}_k^t - \bar{\mathbf{x}}^t) - (\mathbf{x}_l^{t-\tau} - \bar{\mathbf{x}}^{t-\tau})\right\|} \quad , \tag{10}$$

with $\bar{\mathbf{x}}^t$ (resp. $\bar{\mathbf{x}}^{t-\tau}$) being the gravity center of the object $\mathbf{S}^t$ (resp. $\mathbf{S}^{t-\tau}$). In order to facilitate the notation, we will omit the use of indexes $t, t-\tau$ and from now on we will refer to $\mathbf{d}_{ij}^{\tau}$. We should point out that such model does not account only for the motion dynamics of the same landmark (simple case $\mathbf{d}_{ii}^{\tau}$), but for the relative dynamics of the different landmarks. We should also note that this temporal model is of rank τ, and therefore can encode motion dynamics from order 1 up to τ. Last, but not least we would like to mention that this model does not encode acceleration but only relative motions in time.

Like in the static shape prior case, let us consider the empirical distributions of the above mentioned distances, or $p_{ij}^t \equiv p(\mathbf{x}_i^{t0}, \mathbf{x}_j^{t0-t}) \equiv p(\mathbf{d}_{ij}^t)$, for $t \in [1, \tau]$. Like in the former case, we model these densities either with Gaussian mixture models, or non-parametric kernel-based approximations.One issue still to be addressed is the redundancy of the model. This can be done in a similar way than in Sec. 2.2. The set of pairs that are kept between the shapes $\mathbf{S}^t$ and $\mathbf{S}^{t-\tau}$ will be noted $\mathbf{P}_{tmp}^{\tau}$, and the whole

set of temporal pairs will be noted $\mathbf{P}_{tmp}$. We have now built a dynamic shape model $\mathbf{M} = (\mathbf{S}, \mathbf{P}, \mathbf{P}_{tmp})$.

3 Inference Procedure

The main challenges of knowledge-based segmentation and tracking are: (i) appropriate modeling of shape variations and dynamics, (ii) successful inference between the image and the manifold.

Let us consider the simplest possible scenario that aims to detect an object of particular interest from the background in an image $\mathcal{I}$. We formulate the segmentation problem as an energy minimization problem. First, we introduce our model in the image at an initial position and state (it will be noted $\mathbf{M}^0$). Then, we search the optimal displacements $\vec{\mathbf{D}} = (\vec{\mathbf{d}}_1, \ldots, \vec{\mathbf{d}}_n)$ of the control points that give the best compromise between the pairwise prior constraints, encoded in our shape model $\mathbf{M}$, and the fidelity to the image information. Formally, the segmentation of the image $\mathcal{I}$ using the shape model $\mathbf{M}$ is given by:

$$\left(\vec{\mathbf{d}}_1, \ldots, \vec{\mathbf{d}}_n\right) = \underset{\vec{d}_i}{\operatorname{argmin}}\, E\left(\mathbf{M}^0, \mathcal{I}, \left(\vec{d}_1, \ldots, \vec{d}_n\right)\right) \quad . \tag{11}$$

The energy $E(\mathbf{M}^0, \mathcal{I}, \vec{D})$ of displacing the model in the image by the displacement vectors $\vec{D} = (\vec{d}_1, \ldots, \vec{d}_n)$ is the sum of a data-related term $V(\mathbf{S}^0 + \vec{D}, \mathcal{I})$ expressing the image cost of displacing the control points in $\mathbf{S}^0$ from their initial position, and a prior term $V(\mathbf{P}, \vec{D})$ expressing the cost of deforming the pairs in $\mathbf{P}$ from their initial position according to the displacement vectors and with respect to the prior learned distributions :

$$E\left(\mathbf{M}^0, \mathcal{I}, \vec{D}\right) = V\left(\mathbf{S}^0 + \vec{D}, \mathcal{I}\right) + V\left(\mathbf{P}, \vec{D}\right) \quad . \tag{12}$$

Similarly, the tracking problem can be considered as successive segmentation problems where in each image frame $\mathcal{I}_t$, the initialization $\mathbf{M}^t$ corresponds to the result of the previous frame segmentation, and the energy (12) is minimized, with a few adjustments. An additional term $V\left(\mathbf{P}_{tmp}, \vec{D}\right)$ is introduced to account for the dynamic inter-frames dependencies expressing the cost of deforming the pairs in $\mathbf{P}_{tmp}$ from their initial positions, or:

$$E\left(\mathbf{M}^t, \mathcal{I}_t, \vec{D}\right) = V\left(\mathbf{S}^t + \vec{D}, \mathcal{I}_t\right) + V\left(\mathbf{P}, \vec{D}\right) + V\left(\mathbf{P}_{tmp}, \vec{D}\right) \quad . \tag{13}$$

We explain in this section how we define these energy terms, and we detail in particular the relationships between the control points and the image domain. We also develop an optimization procedure that enables to solve (11) using an efficient discrete optimization algorithm.

3.1 Regional Statistics & Image Segmentation

By applying the data-related cost, we seek the optimal separation of the object from the background in terms of visual properties. Let p_{obj} and p_{bcg} be the conditional densities for these two hypotheses. Given that the control points $\mathbf{S}^0 + \vec{D} = \{\mathbf{x}_1^0 + \vec{d}_1, \ldots, \mathbf{x}_n^0 + \vec{d}_n\}$ form a closed boundary, they partition the image domain Ω into an object domain

Ω_{obj} and a background domain Ω_{bcg}. To simplify the notation here, we will refer to a current configuration of the control points that will be noted $\mathbf{S}$. Then by considering the $-\log$ of the posterior probabilities, we express the cost $V(\mathbf{S},\mathcal{I})$ using the regional statistics [44] as follows:

$$V(\mathbf{S},\mathcal{I}) = \sum_{y\in\Omega_{obj}} -\log\left(p_{obj}\left(\mathcal{I}\left(y\right)\right)\right) + \sum_{y\in\Omega_{bcg}} -\log\left(p_{bcg}\left(\mathcal{I}\left(y\right)\right)\right) \quad . \tag{14}$$

In order to evaluate this component and associate it with the proposed shape representation, we decompose the image domain Ω according to the control points $\mathbf{S}$ by considering their Voronoi diagram [Fig. 1(b)][Fig. 3(f)]: $\Omega = \cup_{i=1}^{n}\Omega_i$, where Ω_i is the Voronoi cell associated with the control point $\mathbf{x}_i$. By intersecting these Voronoi cells with the object domain and the background domain, we obtain the partition $\Omega = \cup_{i=1}^{n}(\Omega_{obj,i} \cup \Omega_{bcg,i})$ that relates each pixel of the image to one control point, and specifies its class. Then, one can decompose the global image term (14) into subterms which are defined at the partition cells as follows:

$$V(\mathbf{S},\mathcal{I}) = \sum_{i=1}^{n} V_i(\mathbf{x}_i,\mathcal{I}) \quad , \tag{15}$$

with

$$V_i(\mathbf{x}_i,\mathcal{I}) = \sum_{y\in\Omega_{obj,i}} -\log\left(p_{obj}\left(\mathcal{I}\left(y\right)\right)\right) + \sum_{y\in\Omega_{bcg,i}} -\log\left(p_{bcg}\left(\mathcal{I}\left(y\right)\right)\right) \quad . \tag{16}$$

being the image-related cost associated with the control point position $\mathbf{x}_i$. These terms can be calculated very efficiently per class by combining rasterization techniques and fast integral computing over polygons [39]. We should note that this term uses the entire image domain to determine the image support and can be replaced either using more complex descriptors, or through edge-driven support. In practice, we used simple Gaussian models and mixture of Gaussians models for the object and the background. Such a component will perform well if the data support is strong but will fail to deal with noise, clutter, missing parts, etc. The use of prior knowledge on the expected geometry of the shape could address the above mentioned limitations.

3.2 Image Support & Tracking

In this paragraph, we define relationships between the graph and an image sequence $\{\mathcal{I}_1,\ldots,\mathcal{I}_n\}$, which enables us to infer the optimal positions of the control points to achieve the tracking of the object.

3.2.1 Weak Edges

We define the data-term in an image $\mathcal{I}_t$ of the sequence using its edges. Independently from the previous frames, the optimal positions of the control points should superimpose the interpolated boundary of the object to its edges in the image. A simple way of interpolating the boundary $\mathcal{B}$ is the linear approximation, obtained by connecting successively the control points, one to the other. In order to relate the image to the graph model, we partition $\mathcal{B}$ into boundary segments $\{\mathcal{B}_1,\ldots,\mathcal{B}_n\}$ such that $\mathcal{B}_i$ is the set of the closest boundary points to the control point $\mathbf{x}_i$ ($\mathcal{B} = \cup_{i=1}^{n}\mathcal{B}_i$). Then the optimal

displacements in terms of data terms are obtained by minimizing:

$$\underset{\vec{d}_i}{\operatorname{argmin}} \sum_{k=1}^{n} \int_{\mathcal{B}_k} \mathcal{M}_t(u - \vec{d}_k) du \ , \tag{17}$$

where $\mathcal{M}_t$ is the distance map to the edges of the considered image (we computed in practice the chamfer distance to the canny edges). Although this low-level feature is sensitive to noise, and would often fail, its association with the static prior overcome its usual limitations. The image data can also be useful to inherit temporal consistency to the tracking as explained in the following.

3.2.2 Weak Image Correspondences

Assuming that the tracker is at the frame t of the sequence, we wish to use the previously tracking τ frames in driving the control points to their optimal positions. Then the optimal displacement should lead to the minimum in terms of visual similarity between the current image frame $\mathcal{I}_t$ and the images $\{\mathcal{I}_{t-\tau}, \ldots, \mathcal{I}_{t-1}\}$, or:

$$\underset{\vec{d}_i}{\operatorname{argmin}} \sum_{k=1}^{n} \sum_{s=1}^{\tau} \int_{\mathcal{P}} \left| \mathcal{I}_t(u - \mathbf{x}_i^t - \vec{d}_k) - \mathcal{I}_{t-s}(u - \mathbf{x}_i^{t-s}) \right| du \ , \tag{18}$$

where $\mathcal{P}$ is a patch of a fixed size around the origin. As for the edge-based term, this low-level feature is enhanced thanks to the use of the dynamic prior.
Hence, in the case of tracking, using (17) and (18), we express the cost $V(\mathbf{S}, \mathcal{I})$ as:

$$V(\mathbf{S}, \mathcal{I}_t) = \sum_{i=1}^{n} V_i(\mathbf{x}_i^t, \mathcal{I}_t) \ , \tag{19}$$

with

$$V_i(\mathbf{x}_i^t, \mathcal{I}_t) = \int_{\mathcal{B}_i} \mathcal{M}_t(u - \vec{d}_k) du + \sum_{s=1}^{\tau} \int_{\mathcal{P}} \left| \mathcal{I}_t(u - \mathbf{x}_i^t - \vec{d}_k) - \mathcal{I}_{t-s}(u - \mathbf{x}_i^{t-s}) \right| du \ . \tag{20}$$

3.3 Static Shape Prior Knowledge - Image Segmentation & Object Tracking

In the context of our approach, we have defined the shape model as an incomplete graph. Furthermore, we were able to determine an approximate density of this model using a small number of joint densities. In order to impose the prior, we minimize the cost $V(\mathbf{P}, \vec{D})$ that we decompose over all the pairs:

$$V(\mathbf{P}, \vec{D}) = \alpha \underbrace{\sum_{(i,j) \in \mathbf{P}_l} V_{ij}(\mathbf{x}_i^0 + \vec{d}_i, \mathbf{x}_j^0 + \vec{d}_j)}_{\text{local prior cost}} + \beta \underbrace{\sum_{(i,j) \in \mathbf{P}_g} V_{ij}(\mathbf{x}_i^0 + \vec{d}_i, \mathbf{x}_j^0 + \vec{d}_j)}_{\text{global prior cost}} \ , \tag{21}$$

with

$$V_{ij}(\mathbf{x}_i^0 + \vec{d}_i, \mathbf{x}_j^0 + \vec{d}_j) = -\log\left(p_{ij}(\mathbf{x}_i^0 + \vec{d}_i, \mathbf{x}_j^0 + \vec{d}_j) \right) \ . \tag{22}$$

This model allows for the encoding of global dependencies as local combinations of individual pairwise densities. Some examples of the impact of this term for a random

collection of points with respect to the hand model [Fig. 3(a)] are shown in [Fig. 3(b)-3(e)]. The parameters (α, β) control the relative influence of inter and intra cluster constraints.

3.4 Dynamic Shape Prior Knowledge & Object Tracking

Assuming that the tracker is at the frame t of the sequence, we wish to use the previously tracking τ frames in driving the control points to their optimal positions, by applying the dynamic constraints. In order to impose this prior, we minimize the cost $V\left(\mathbf{P}_{tmp}, \vec{D}\right)$ that we decompose over all the control points $\mathbf{x}_i^t$ at the current frame or:

$$V\left(\mathbf{P}_{tmp}, \vec{D}\right) = \gamma \sum_{i=1}^{n} V_i^t(\mathbf{x}_i^t + \vec{d}_i) \ , \tag{23}$$

with

$$V_i^t(\mathbf{x}_i^t + \vec{d}_i) = \sum_{s=1}^{\tau} \sum_{j=1}^{n} -\log\left(p_{ij}^s(\mathbf{x}_i^t + \vec{d}_i, \mathbf{x}_j^{t-s})\right) \ . \tag{24}$$

3.5 The Designed Energy

One can now integrate the data term with the prior term towards knowledge-based segmentation, by combining (12), (15),and (21):

$$\begin{aligned} E\left(\mathbf{M}^0, \mathcal{I}, \vec{D}\right) = & \sum_{i=1}^{n} V_i(\mathbf{x}_i^0 + \vec{d}_i, \mathcal{I}) + \alpha \sum_{(i,j)\in\mathbf{P}_l} V_{ij}(\mathbf{x}_i^0 + \vec{d}_i, \mathbf{x}_j^0 + \vec{d}_j) \\ & + \beta \sum_{(i,j)\in\mathbf{P}_g} V_{ij}(\mathbf{x}_i^0 + \vec{d}_i, \mathbf{x}_j^0 + \vec{d}_j) \ . \end{aligned} \tag{25}$$

Knowledge-based tracking at the frame t is expressed in a similar manner, by combining (13), (19), (21) and (23):

$$\begin{aligned} E\left(\mathbf{M}^t, \mathcal{I}_t, \vec{D}\right) = & \sum_{i=1}^{n} V_i(\mathbf{x}_i^t + \vec{d}_i, \mathcal{I}_t) + \alpha \sum_{(i,j)\in\mathbf{P}_l} V_{ij}(\mathbf{x}_i^t + \vec{d}_i, \mathbf{x}_j^t + \vec{d}_j) \\ & + \beta \sum_{(i,j)\in\mathbf{P}_g} V_{ij}(\mathbf{x}_i^t + \vec{d}_i, \mathbf{x}_j^t + \vec{d}_j) + \gamma \sum_{i=1}^{n} V_i^t(\mathbf{x}_i^t + \vec{d}_i) \ . \end{aligned} \tag{26}$$

In the following, we will use the same notation for (25) and (26) (where the segmentation case corresponds to $\gamma = 0$) and write:

$$E\left(\vec{D}\right) = \sum_{i=1}^{n} V_i(\vec{d}_i) + \alpha \sum_{(i,j)\in\mathbf{P}_l} V_{ij}(\vec{d}_i, \vec{d}_j) + \beta \sum_{(i,j)\in\mathbf{P}_g} V_{ij}(\vec{d}_i, \vec{d}_j) + \gamma \sum_{i=1}^{n} V_i^t(\vec{d}_i) \ . \tag{27}$$

3.6 The Energy Minimization

The optimization of this cost function (27) in the continuous domain is rather problematic. One can expect that it is not convex and therefore a gradient-based optimization

will fail. In order to optimize such a cost function, we consider recent results from discrete optimization.

We make two assumptions that are most often verified in practice. First, the initial positions of the control points are within the image domain, and that is why we can assume an upper bound on the maximum displacements that would lead to the solution. Second, we consider that the precision required about the solution is specified, which enables to choose a quantization step of the displacement vectors $\vec{D}$. Then, we can approximate the continuous deformations of our shape model towards the solution by a finite set of displacements vectors $\vec{D} = \{\vec{d^1}, \ldots, \vec{d^z}\}$. Let $\mathcal{L} = \{1, \ldots, z\}$ be the set of labels associated the quantization $\vec{D} = \{\vec{d^1}, \ldots, \vec{d^z}\}$ of the displacements. Then, displacing the control point $\mathbf{x}_i$ by the vector $\vec{d^{l_i}}$ is equivalent to assigning the label l_i to $\mathbf{x}_i$, and the minimization of the energy in (27) can be written as a labeling problem, or:

$$(\mathbf{l}_1, \ldots, \mathbf{l}_n) = \underset{l_i \in \mathcal{L}}{\operatorname{argmin}} \, E\left(l_1, \ldots, l_n\right) \quad , \tag{28}$$

with

$$E\left(l_1, \ldots, l_n\right) = \sum_{i=1}^{n} V_i(l_i) + \alpha \sum_{(i,j)\in\mathbf{P}_l} V_{ij}(l_i, l_j) + \beta \sum_{(i,j)\in\mathbf{P}_g} V_{ij}(l_i, l_j) + \gamma \sum_{i=1}^{n} V_i^t(l_i) \quad , \tag{29}$$

where $V_i(\mathbf{x}_i, l_i) = V_i(\mathbf{x}_i + \vec{d_i}, \mathcal{I})$, $V_{ij}(\mathbf{x}_i, \mathbf{x}_j, l_i, l_j) = V_{ij}(\mathbf{x}_i + \vec{d_i}, \mathbf{x}_j + \vec{d_j})$ and $V_i^t(l_i) = V_i^t(\mathbf{x}_i + \vec{d_i})$. In such a context, the problem of finding the most appropriate deformation of the initial shape can be expressed using an MRF with singleton and pairwise interactions between the control points. We should note that such an approach is invariant to translation, rotation and scale (due to the definition of (1)). Recovering the optimal solution of this objective function is known to be an NP-hard problem and the complexity is influenced mostly from the pairwise potentials function. Hence, we consider an approximate solution to the labeling problem using the Primal-Dual algorithm [25].

The cardinality of the label set is quite important since on one hand it defines the accuracy of the search, while on the other hand increases the complexity of the algorithm. In order to address the above mentioned issues, first we consider an approach that is incremental in terms of displacements while reducing the number of interactions between the nodes of the graph, and retaining the ability to encode the global structure. To this end, we cope with the accuracy issue, that is closely related to the quantization of $\vec{D}$, by using a pyramidal coarse-to-fine approach. Each level of the pyramid corresponds to a quantization step that is refined in the following level. To speed up the convergence in each level of the pyramid, we also adopt an incremental approach in terms of the label set [41], where in each iteration κ we look for the set of labels that will improve the current solution by minimizing:

$$\begin{aligned} E^{\kappa}\left(l_1, \ldots, l_n\right) = & \sum_{i=1}^{n} V_i(\mathbf{x}_i(\kappa), l_i) + \alpha \sum_{(i,j)\in\mathbf{P}_l} V_{ij}(\mathbf{x}_i(\kappa), \mathbf{x}_j(\kappa), l_i, l_j) \\ & + \beta \sum_{(i,j)\in\mathbf{P}_g} V_{ij}(\mathbf{x}_i(\kappa), \mathbf{x}_j(\kappa), l_i, l_j) + \gamma \sum_{i=1}^{n} V_i^t(\mathbf{x}_i(\kappa) + \vec{d_i}) \quad , \end{aligned} \tag{30}$$

$$\text{with } \mathbf{x}_i(\kappa) = \mathbf{x}_i + \sum_{k=1}^{\kappa-1} \vec{d}^{\mathbf{l}_i(k)} \quad , \tag{31}$$

with $\mathbf{l}_i(k)$ being the optimal label associated with the i^{th} control point at iteration k. Towards computational efficiency and localization of a good minimum, we adopt a fast and efficient method for the optimization, the Primal-Dual algorithm [25] that is based on linear programming and takes benefit of the duality theorem. The main challenge of optimizing the above objective function relates with the fact the we have arbitrary pairwise potentials. Therefore the use of method like graph-cuts [5] is prohibited while at the same time the use of more advanced optimization like belief-propagation networks [43] is also problematic due to the structure of the graph.

4 Experimental Validation

In order to validate the performance of our method, we considered different applications and experimental settings. We present here our results for the modeling and the segmentation of the hand and then for the left ventricle in CT images. Next, we show tracking results for walking people sequences.

4.1 Segmentation of the Hand

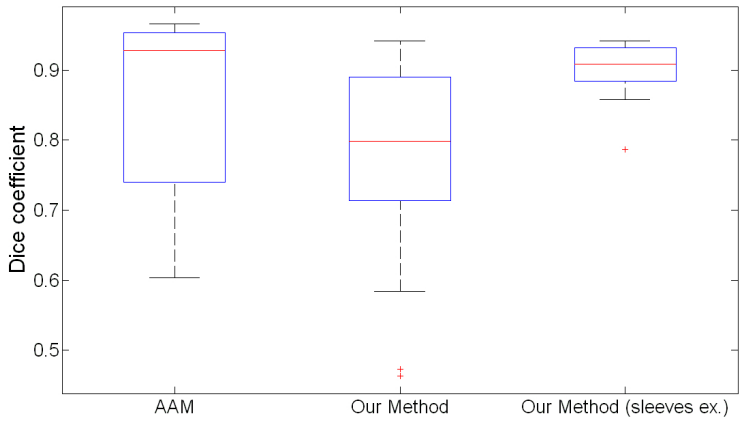
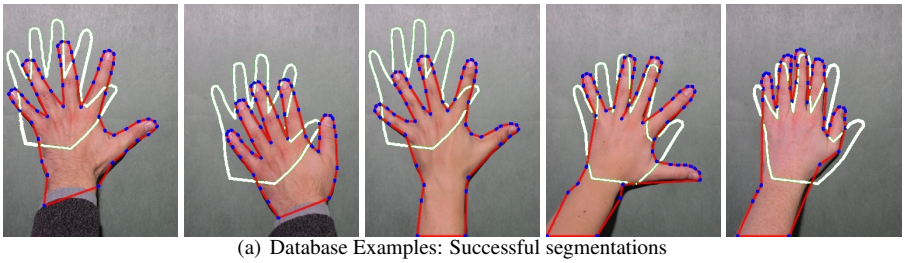
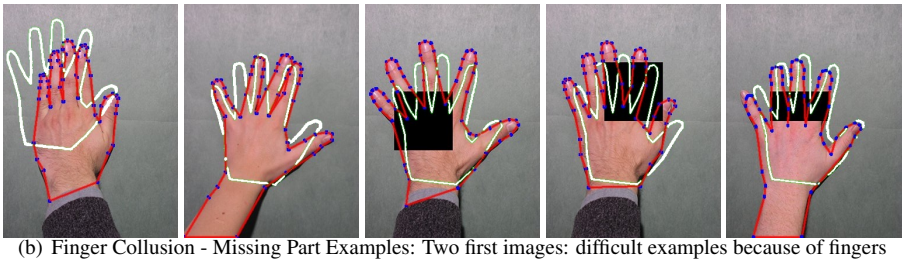


Figure 4: Comparison between our method and AAM.

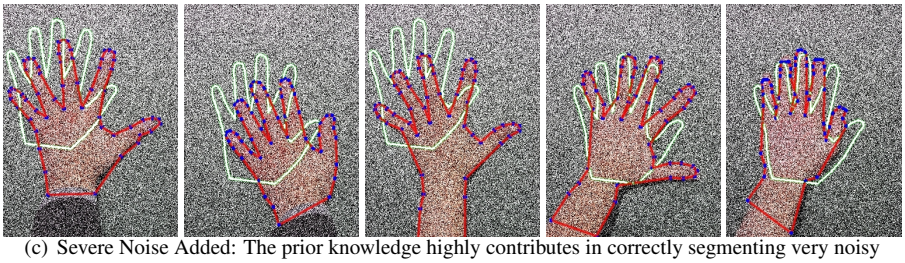
We considered the application of modeling the hand using a 2D 40-example dataset of annotated left hands, showing different relative finger positions, hand sizes, and texture [35]. On each hand contour, 56 landmarks were used to describe the structure. After the global alignment of the examples, we have performed clustering on the distribution space as described in section 2.3, using the shape map [26]. The clustering provided 11 clusters shown in [Fig. 3(a)]. The constructed model was used as a shape constraint as shown in [Fig. 3(b)-3(e)], and applied in different segmentation settings. We considered a multi-scale implementation of the approach using gradually an increasing number of control points to accelerate convergence. First, we segmented correctly 37 out of the 40 examples of the database. Examples of the results we obtained are shown in [Fig. 5(a)]. We also compared quantitatively our method to AAM segmentation [Fig. 4]. We can see in this figure that our algorithm performs better



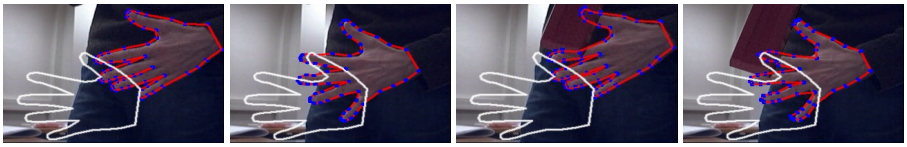
(a) Database Examples: Successful segmentations



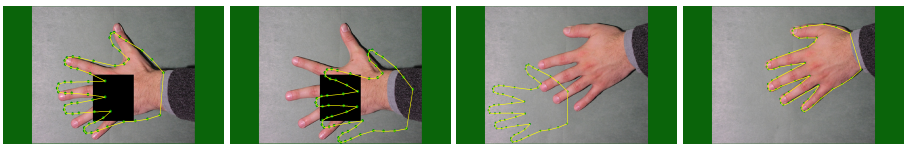
(b) Finger Collusion - Missing Part Examples: Two first images: difficult examples because of fingers collusions. Three last images: segmentation of hands with missing parts.



(c) Severe Noise Added: The prior knowledge highly contributes in correctly segmenting very noisy images.



(d) Video Frames - Partial Occlusions: Real video frames: cluttered background and occlusions.



(e) AAM results: succeeds with the learning examples but fails with occlusions. Initialization on the left - result on the right.

Figure 5: Model-based segmentation of the hand. Initialization is shown in white, segmentation in red, and the final control points positions in blue.

quantitatively with examples where the arm is hidden by a sleeve. In the case of nude arms, the data term drives the model to "oversegment" the hand in comparison with the ground truth, which explains our results. These "oversegmentations" are visually correct (especially the fingers are correctly segmented) as we can see in [Fig. 5(a)]. The

three examples where our method did not succeed are particularly difficult because they exhibit occlusion between the fingers, which can cause folding in the evolving surface. Towards checking the robustness of the method, we removed some hands parts for several examples, and despite the important missing structure, the results were quite satisfactory as shown in [Fig. 5(b)]. The prior weight in these cases was increased, and enforced the correct segmentation, as the data term was less reliable. Furthermore, to validate the robustness of our method, we added severe Gaussian noise to the database images. The segmentations obtained in [Fig. 5(a)] are completely or almost recovered, thanks to the prior knowledge, as it is shown in [Fig. 5(c)]. Eventually, we used our segmentation method in a real setting, on hand video frames, with a cluttered background and partial occlusion cases. [Fig. 5(d)] gives some examples of the obtained segmentations. We could reproduce the result we obtained on the noisy images using AAM segmentation [11], but this algorithm could not reproduce our results for the occlusion cases. In our experiments, one iteration lasts approximately 1s using a non-optimized program, on a DELL Duo Computer (3GHz, 3GB).

4.2 Segmentation of the Left Ventricle in CT images

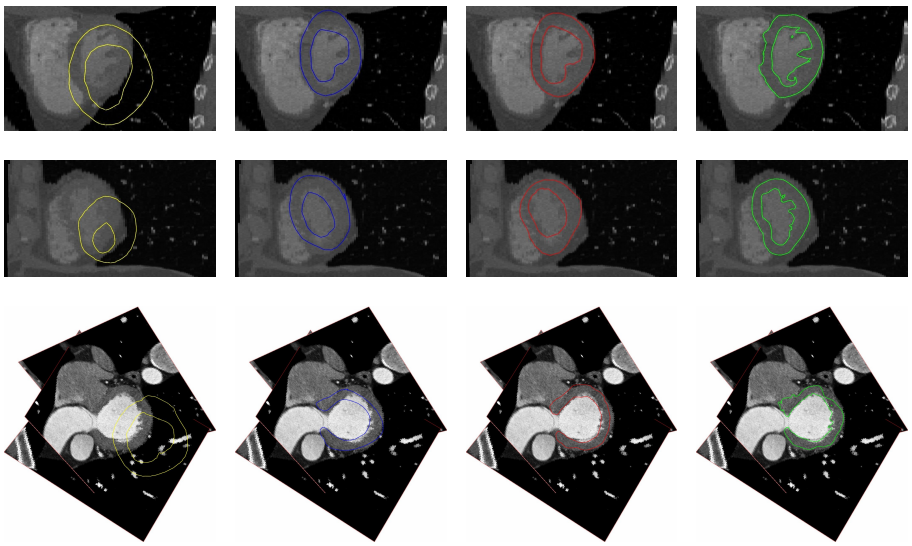


Figure 6: Segmentation results: 3 “unseen” examples (do not belong to the training set). Initialization in yellow, shape after affine transform in blue, final segmentation after TPS deformation in red. Random Walker result in green.

We used a dataset of 28 3D CT images, having an approximate mean size of $512 \times 512 \times 250$ voxels, where the voxels size is about $0.35 \times 0.35 \times 0.35$mm. While no expert manual segmentation was available for this dataset, we could compare to the results given by the Random Walker algorithm described in [19]. First, to build the model, we selected randomly 11 CT images as a training set. We placed manually on the surface of the left ventricle of one example from the training set 90 control points, that we will call P_{90}. The remaining 10 examples of the training set where then registered to the labeled example using the method described in [36] and correspon-

Table 1: Comparison of our method with expert segmentation

Correctly Segmented Voxels (True Positives)	False Positives	Seg. Time (3GHz,3GB)
85.12%±7.3	15.3%±10.2	90 s±10

dences between the control points instances were consequently deduced. We learned next the probability density distributions p_{ij} of the normalized chord lengths (Sec. 2.1) and p_{obj} and p_{bck} of the regions grey levels (Sec. 3.1) as Gaussian distributions, using the training set. Moreover, after smoothing the segmentation mask obtained by [19] on the labeled example, we generated a meshed surface $\mathcal{S}$ of the myocardium and the blood pool. By intersecting $\mathcal{S}$ with the voronoi diagram of the set of control points P_{90} we obtained the classes cells Ω^i_{obj} and Ω^i_{bck} (Sec. 3.1). In particular, 4 of the 90 control points are interesting as one is located in the apical area, and the others in the basal area (this set will be called P_4). Figure 1 shows the obtained surface $\mathcal{S}$ with the control points P_{90} and P_4, and the voronoi cells Ω_{obj} and Ω_{bck} of the apical control point. The results presented in this part were obtained using an incomplete graph where every control point was paired with its 10 farthest neighbors (the clustering was not considered).

As a first application, we tested the consistency of the learned prior by only minimizing the shape term of the designed energy, and canceling the data term, starting from randomly perturbed positions of the control points. The prior constraints made the initial shape converge in all cases to the mean shape. Next, the segmentation was performed by minimizing the energy (28) using the schema (30) by following two steps: (i) the surface $\mathcal{S}$ is deformed from its initial position by using the control points P_4, and by applying an affine transformation to the mesh after each iteration. After the convergence of this first step, (ii) the control points P_{90} are introduced and the mesh is transformed using a Thin Plate Spline (TPS) [4] deformation. For the segmentation experiments, the data slices were resized to 128×128 pixels.

We compared our results quantitatively for the whole dataset to [19] and compiled them in Tab. 1. The differences between the two methods explain these quantitative results, as our segmentation is smooth whereas the output of [19] is rather noisy as we can see in [Fig. 6].

(a) frame 0 (b) frame 66 (c) frame 94

Figure 7: Frames extracted from a tracking sequence using our method with the static prior. Note that the tracking quality is better when using the dynamic prior for the frame 66 for example (see [Fig. 8(b)]): here one leg is missed.

4.3 Tracking of Walking People

We considered a commonly-used example in tracking: walking people. This case shows the generality of our method for articulate objects, which is not specifically optimized for this particular experiment. The problem of tracking walking people provides a deformable object with interesting dynamics. We used in our experiments video sequences from the Georgia Institute of Technology database (`http://www.cc.gatech.edu/cpl/projects/hid/Description.html`). The results were run on a PC equipped with an Intel Pentium M 2.0GHz processor and 1.5GB memory. The tracking lasts approximately 1 second per frame using a non-optimized code. The results presented in this part were obtained using a complete graph in space and time (the clustering was not considered).

We first selected a total of 455 frames of walking persons from the database with different gaits. We labeled manually these frames placing the landmarks at corresponding positions. Then we used these labeled images to learn a static prior and a dynamic prior. Next,for the testing, we applied our trained priors to three video sequences. The model points were initialized close to the walking target. For the qualitative evaluation, we compare the results we obtain using the static prior, and then by using the static and the dynamic prior. From our experiment we observe that the results with dynamic priors introduce less flips, and have a better quality than those from the static prior tests. Figures 7 and 8 show examples of the obtained results. Although the features we use are weak, our algorithm is able to track the object due to the learned prior.

To evaluate the results quantitatively, we tested our algorithm with the labeled frames. We measure the average distance of the computed result to the ground truth in pixel unit. These measures are performed for the three sequences, to compare the performance of the static prior to the dynamic prior. We also reproduce the same experiments by adding a pre-processing that consists in subtracting the background. These tests aim to evaluate the robustness of the tracker to the background noise. The results of these experiments are summarized in Table 2. We conclude from these results that

(a) frame 1 (b) frame 66 (c) frame 94

(d) frame 120 (e) frame 163 (f) frame 183

Figure 8: Frames extracted from a tracking sequence using our method with the dynamic prior. Note that the tracking quality is better than in the static prior case for the frame 66 for example (see [Fig. 7(b)]).

Table 2: Comparison between the dynamic prior and the static prior, in terms of average distance to the ground truth in pixels (mean±std). The mention "-Bck." corresponds to a background subtraction preprocessing.

Sequence #	1	2	3
Static	3.25±1.40	3.97±0.96	2.52±0.47
Static - Bck.	2.06±0.53	3.30±0.34	2.52±0.43
Dynamic	2.51±0.88	3.39±0.37	2.42±0.31
Dynamic - Bck.	2.39±0.53	3.31±0.59	2.40±0.47

the dynamic prior outperforms the static one. Moreover, its use increases the robustness to the edge detection noise.

5 Discussion

In this report we have proposed a novel approach to knowledge-based segmentation and tracking. Our main contribution consists of modeling the co-dependencies between control points deformations in space and time, towards a compact, sparse but efficient shape representation using an incomplete graph that was determined through an unsupervised clustering approach on the relevance of statistical behavior of control points deformations. This representation is combined with a data term like regional statistics or edge-based costs in order to perform inference of the model location or analogously segmentation in new image data (or tracking in a new image frame). To this end, a MRF is considered where singleton potentials account for the image support, and for the dynamic prior in the tracking case, while pair-wise potentials encode the shape prior. Our approach can claim certain optimality properties thanks to the efficient linear programming optimization techniques considered in this work. Furthermore, our approach can make full use of the regional statistics and the obtained minimum is in such a case the one corresponding to the entire image potential.

The proposed method learns the structure and local deformation statistics of an object from a set of training examples. The structure is used to construct a compact graph, which represents the mutual dependencies in the training data in an efficient manner. Based on this graph, the pair-wise shape statistics, and local appearance information, a MRF is created, which captures the training set behavior in a compact representation. The search is formulated as an iterative labeling problem during which positions in the search image are assigned to the control points. The compact structure allows for an efficient solution of the graph labeling problem during search. In terms of clustering, the distance between observations has a critical impact and should be further investigated. The temporal aspect of priors is also something of great importance, and we show its potential in human tracking. Extending the current framework to the temporal domain in medical imaging towards 4D segmentation is a direction that should be investigated.

References

[1] A. Andreopoulos and J. K. Tsotsos. Efficient and generalizable statistical models of shape and appearance for analysis of cardiac mri. *Medical Image Analysis*, 12(3):335–357, June 2008.

[2] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[3] M. Black and A. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 1998.

[4] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989.

[5] Y. Boykov and O. Veksler. *Handbook of Mathematical Models in Computer Vision*, chapter Graph Cuts in Vision and Graphics: Theories and Applications. Springer Verlag, 2006.

[6] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[7] Y. Chen, F. Huang, H. D. Tagare, M. Rao, D. Wilson, and E. A. Geiser. Using prior shape and intensity profile in medical image segmentation. In *ICCV '03: Proceedings of the 9th IEEE International Conference on Computer Vision*, volume 2, pages 1117 – 1124, 2003.

[8] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, July 2006.

[9] R. T. Collins. Mean-shift blob tracking through scale space. In *CVPR '03: Proceedings of the 2003 Conference on Computer Vision and Pattern Recognition*, volume 2, pages 234–240, 2003.

[10] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[11] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[12] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[13] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *CVPR '05: Proceedings of the 2005 Conference on Computer Vision and Pattern Recognition*, pages 10–17, 2005.

[14] D. Cremers, T. Kohlberger, and C. Schnörr. Nonlinear shape statistics in Mumford–Shah based segmentation. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision*, pages 93–108, 2002.

[15] P. F. Felzenszwalb. Representation and detection of deformable shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):208–220, 2005.

[16] A. Frangi, W. Niessen, and M. Viergever. Three-dimensional modeling for functional analysis of cardiac images: A review. *IEEE T. Med. Imaging (TMI)*, 20(1):2–5, January 2001.

[17] D. Freedman and P. Drineas. Energy minimization via graph cuts: Settling what is possible. In *CVPR '05: Proceedings of the 2005 Conference on Computer Vision and Pattern Recognition*, pages 939–946, Washington, DC, USA, 2005. IEEE Computer Society.

[18] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[19] L. Grady, V. Sun, and J. Williams. *Mathematical Models of Computer Vision: The Handbook*, chapter Interactive Graph-Based Segmentation Methods in Cardiovascular Imaging, pages 453–469. Springer Verlag, 2005.

[20] L. Gu, E. Xing, and T. Kanade. Learning gmrf structures for spatial priors. In *CVPR '07: Proceedings of the 2007 Conference on Computer Vision and Pattern Recognition*, pages 1–6, 2007.

[21] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[22] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[23] P. Kohli, J. Rihan, M. Bray, and P. H. Torr. Simultaneous segmentation and pose estimation of humans using dynamic graph cuts. *International Journal of Computer Vision*, 79:285298, 2008.

[24] N. Komodakis. Clustering via lp-based stabilities. In *NIPS '08: Advances in Neural Information Processing Systems 21*, 2008. submitted to NIPS.

[25] N. Komodakis, G. Tziritas, and N. Paragios. Performance vs computational efficiency for optimizing single and dynamic mrfs: Setting the state of the art with primal dual strategies. *Computer Vision and Image Understanding*, 112(1):14–29, 2008.

[26] G. Langs and N. Paragios. Modeling the structure of multivariate manifolds: Shape maps. In *CVPR '08: Proceedings of the 2008 Conference on Computer Vision and Pattern Recognition*, 2008.

[27] B. Lucas and T. Kanade. *Detection and Tracking of Point Features*. Carnegie Mellon Univ., Tech. Rep. CMU-CS-91-132, 1991.

[28] T. McInerney and D. Terzopoulos. Deformable models in medical images analysis: a survey. *Medical Image Analysis*, 1(2):91–108, 1996.

[29] S. Mitchell, B. Lelieveldt, R. van der Geest, H. Bosch, J. Reiver, and M. Sonka. Multistage hybrid active appearance model matching: segmentation of left and right ventricles in cardiac mr images. *IEEE Transactions on Medical Imaging*, 20(5):415–423, May 2001.

[30] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[31] N. Paragios, M. Rousson, and V. Ramesh. Matching distance functions: A shape-to-area variational approach for global-to-local registration. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision*, volume II, pages 775–789, London, UK, 2002. Springer-Verlag.

[32] M. Rousson and N. Paragios. Prior knowledge, level set representations & visual grouping. *International Journal of Computer Vision*, 76(3):231–243, 2008.

[33] T. Schoenemann and D. Cremers. Globally optimal image segmentation with an elastic shape prior. In *ICCV '07: Proceedings of the 11th IEEE International Conference on Computer Vision*, pages 1–6, 2007.

[34] J. Shi and C. Tomasi. Good features to track. In *CVPR '94: Proceedings of the 1994 Conference on Computer Vision and Pattern Recognition*, 1994.

[35] M. B. Stegmann and D. D. Gomez. A brief introduction to statistical shape analysis, 2002.

[36] M. Taron, N. Paragios, and M.-P. Jolly. Registration with uncertainties and statistical modeling of shapes with variable metric kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):99–113, 2009.

[37] C. Taylor and D. Cooper. Shape verification using belief updating. In *BMVC '90: Proceedings of the British Machine Vision Conference*, 1990.

[38] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: recovering 3d shape and nongrid motion. *Artificial Intelligence*, 36(1):91–123, 1988.

[39] G. B. Unal, A. J. Yezzi, and H. Krim. Information-theoretic active polygons for unsupervised texture segmentation. *International Journal of Computer Vision*, 62(3):199–220, 2004.

[40] R. Urtasun, D. J. Fleet, and P. Fua. Motion models for 3d people tracking. *Computer Vision and Image Understanding*, 2-3(104):157–177, 2006.

[41] O. Veksler. *Efficient graph-based energy minimization methods in computer vision*. PhD thesis, Graduate School of Cornell University, 1999.

[42] J. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(30):283–298, 2008.

[43] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7):2282–2312, 2005.

[44] S. C. Zhu and A. L. Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):884–900, 1996.

Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 Orsay Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399